Edge Impulse Wake Word Model Training | Tech Roundup

Today, we will walk you step by step through learning **how to train a voice keyword model and deploy it to embedded hardware, using the Edgi-Talk platform to adapt Edge Impulse,** Of course, the principles also apply generally to other ARM embedded platforms. Let's see how to get Edgi-Talk started with edge machine learning!

Edge Impulse Introduction

Edge Impulse is the leading-edge AI development platform, dedicated to helping developers, engineers, and enterprises quickly deploy smart AI to any edge device (such as microcontrollers, sensors, and IoT hardware).

You can easily **collect sensor data, build datasets, train and optimize machine learning models, then deploy with one click to thousands of hardware ecosystems worldwide,** without needing deep AI expertise to achieve rapid prototyping and commercialization.

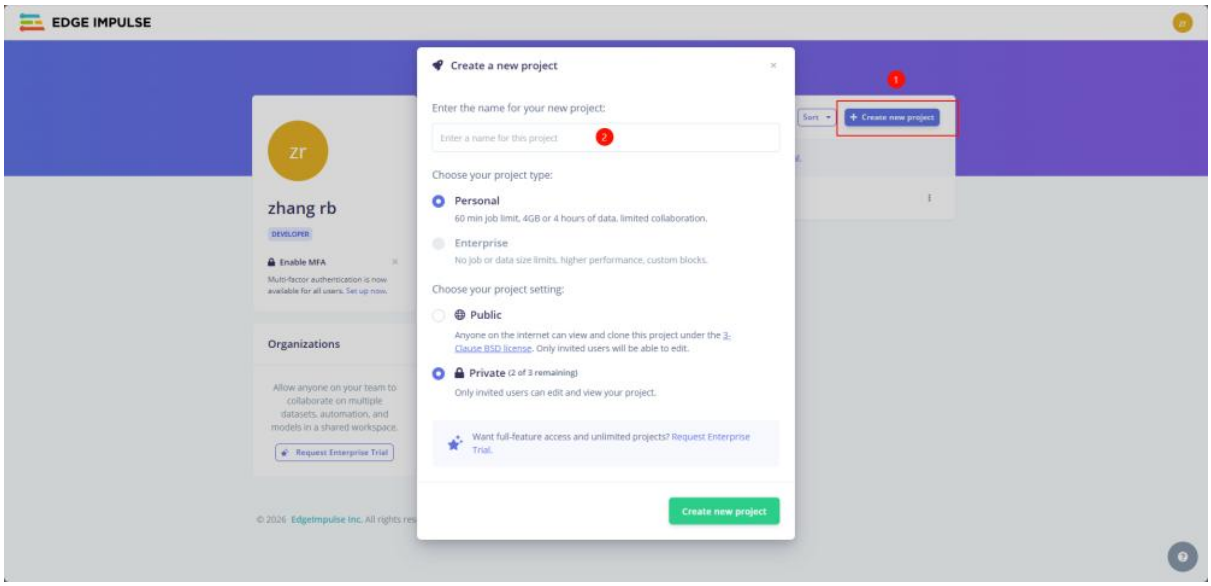
1 Create an Account

Steps:

1. Log in to Edge Impulse, https://edgeimpulse.com/

2. Register an account;

3. Via create new project create a project



2 Record Dataset

1.Flash Edgi-Talk dev board with the UAC firmware, used for collecting audio data: uac-firmware.hex; (**Link:** https://github.com/RT-Thread-Studio/sdk-bsp-psoc_e84-edgi-talk/releases/download/1.1.0/uac-firmware.hex)

Steps to flash firmware via ModusToolbox Programmer

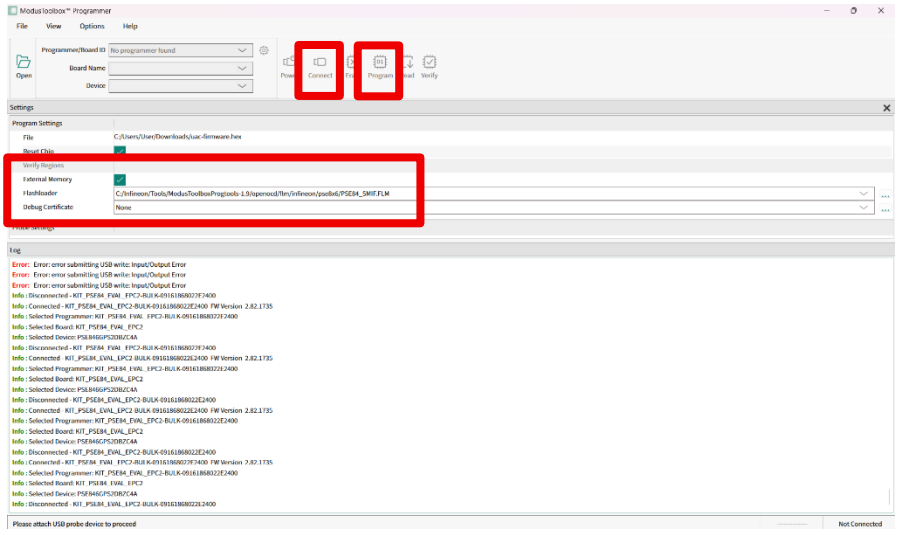
1. **Launch the app**

"C:\Infineon\Tools\ModusToolboxProgtools-1.9\mtb-programmer\mtb-programmer.exe"

2. **Connect the board** — plug the USB cable into the **DAP** port (this is what powers the board and lets the programmer talk to it).
3. **Check the top settings bar** — it should auto-populate:
    - Programmer/Board ID: KIT_PSE84_EVAL_EPC2-BULK-...
    - Board Name: KIT_PSE84_EVAL_EPC2
    - Device: PSE846GPS2DBZC4A
4. **Click "Connect"** (top toolbar) to establish the connection to the board.
5. **Open the Settings panel** (should show automatically, or click the gear icon) and set:
    - **File** → browse to your .hex firmware file (e.g. C:\Users\User\Downloads\uac-firmware.hex)
    - **Reset Chip** → ✅ ticked
    - **External Memory** → ✅ **ticked (this was the critical step)** — without this, the tool won't set up the external QSPI flash region and the write will silently fail with "wrote 0 bytes"
    - **Flashloader** → leave as "Default" (it auto-selects the right one once External Memory is ticked)
6. **Click "Program"** in the top toolbar.
7. **Check the Log panel at the bottom** for success — look for:
    - Info : flash 'cmsis_flash' found at 0x60000000 (confirms the QSPI bank was set up)
    - Info : wrote XXXXX bytes from file ... where XXXXX is a real nonzero number

- Info : ** Programming Finished **

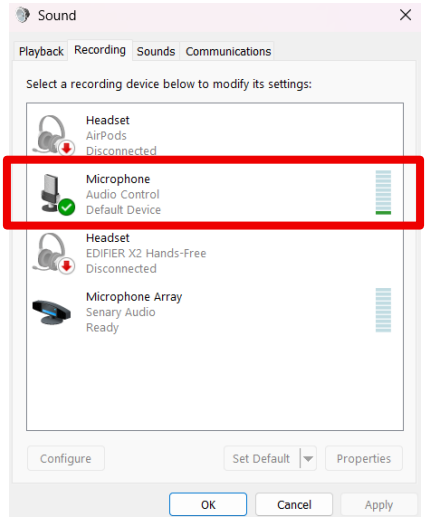
8. **Unplug the DAP cable, plug the cable into the "USB" port instead** — this is the board's application-side data port, separate from the programmer port, and it's what shows up as a working audio device to your PC once the new firmware is running.



2. **Verifying the board shows up as a microphone**

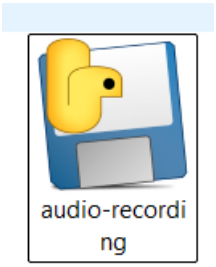
1. **Make sure the cable is plugged into the "USB" port** on the board (not DAP) — this only works with the application-side port connected.
2. Open **Sound settings** via Control Panel:
    - Right-click the speaker icon in your Windows taskbar → **Sound settings** (or **Sounds**), OR
    - Open **Control Panel** → **Sound**
3. Click the **"Recording"** tab.

4. Look for a new entry: For example, Microphone



Confirm it is picking up audio: with that device selected, speak or make noise near the board — you should see the small level meter (the vertical bars next to it) move/light up in response, confirming live audio input.

3.Download and open the audio dataset recording software: Audio-recording.exe; (**Link:** https://github.com/RT-Thread-Studio/sdk-bsp-psoc_e84-edgi-talk/releases/download/1.1.0/audio-recording.exe)



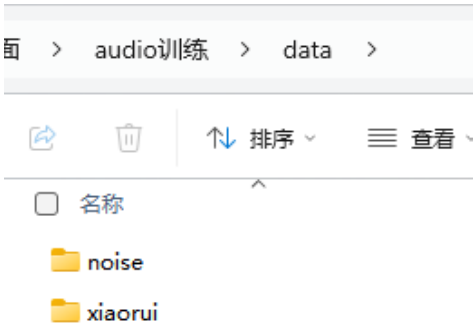
4.For the audio input device, select: Audio Control (MME) device;

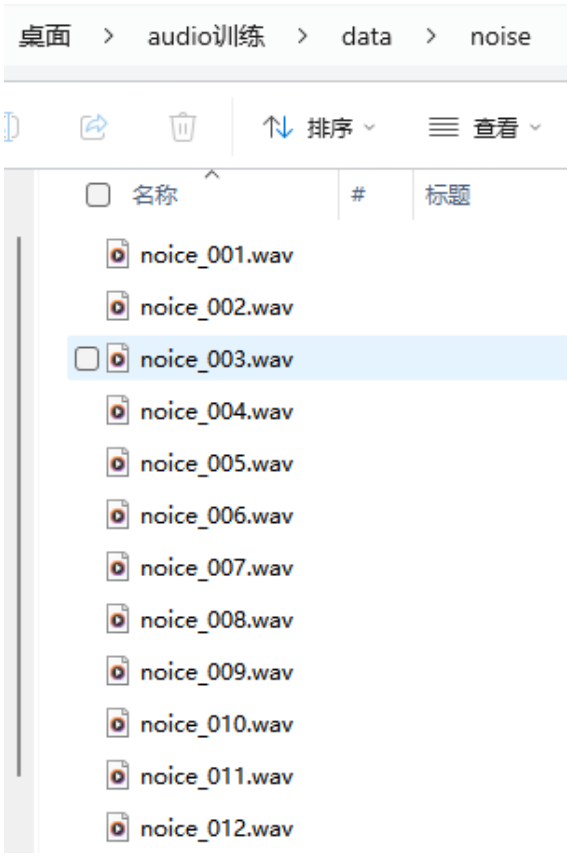
5.Click Start Recording to make one recording lasting up to 10 seconds; ensure clear speech during recording — wake word recordings need pauses of 1-2s between repetitions;

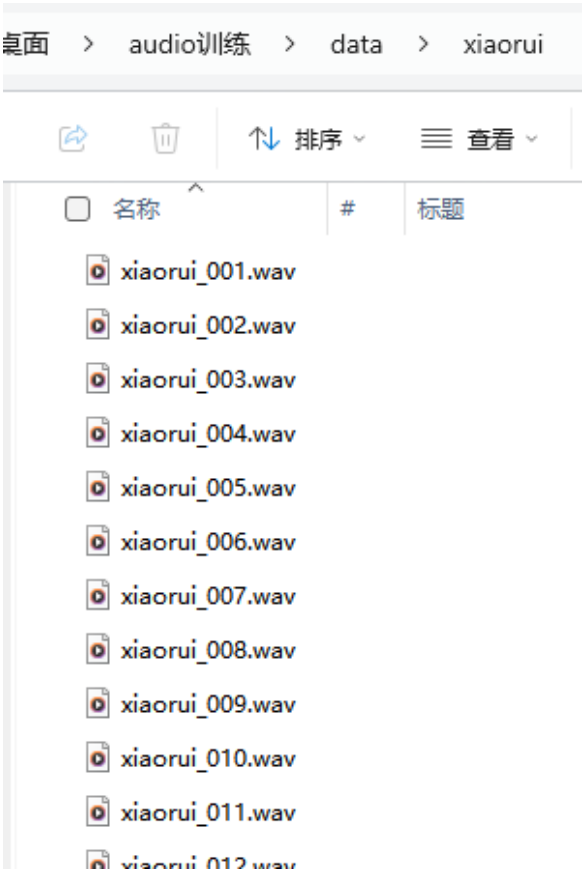
6.After completing one recording, click on the right side "Save Recording" option to save it;

7.The current sequence number will increment by default; repeat the above steps to record 20-50 keyword audio files;

8.Ultimately, record two types of audio files: one for the wake word and one for noise;


面 > audio训练 > data >
排序
查看
名称
noise
xiaorui

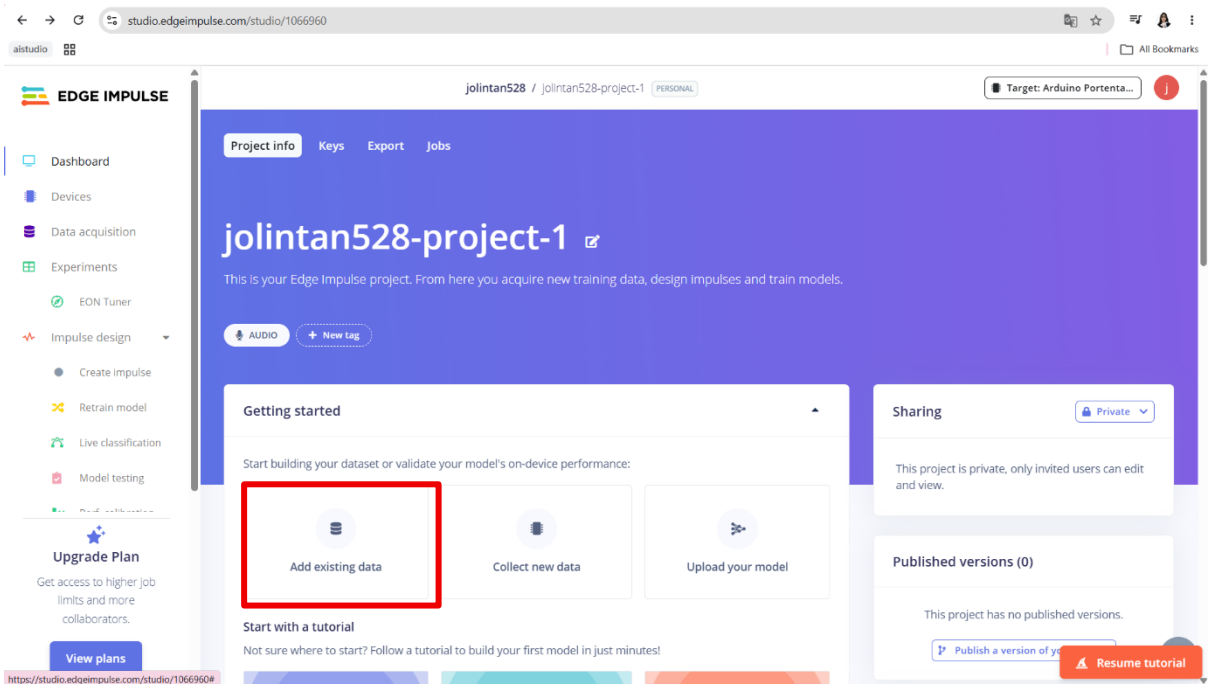

桌面 > audio训练 > data > noise
排序
查看
名称
#
标题
noice_001.wav
noice_002.wav
noice_003.wav
noice_004.wav
noice_005.wav
noice_006.wav
noice_007.wav
noice_008.wav
noice_009.wav
noice_010.wav
noice_011.wav
noice_012.wav

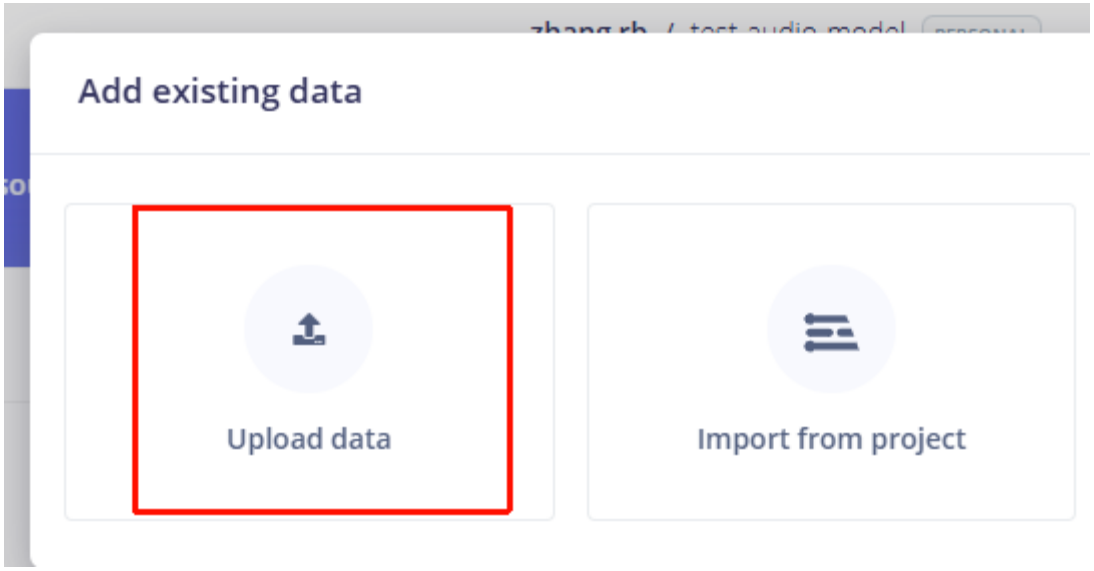

桌面 > audio训练 > data > xiaorui
排序
查看
名称
#
标题
xiaorui_001.wav
xiaorui_002.wav
xiaorui_003.wav
xiaorui_004.wav
xiaorui_005.wav
xiaorui_006.wav
xiaorui_007.wav
xiaorui_008.wav
xiaorui_009.wav
xiaorui_010.wav
xiaorui_011.wav

3 Data Upload

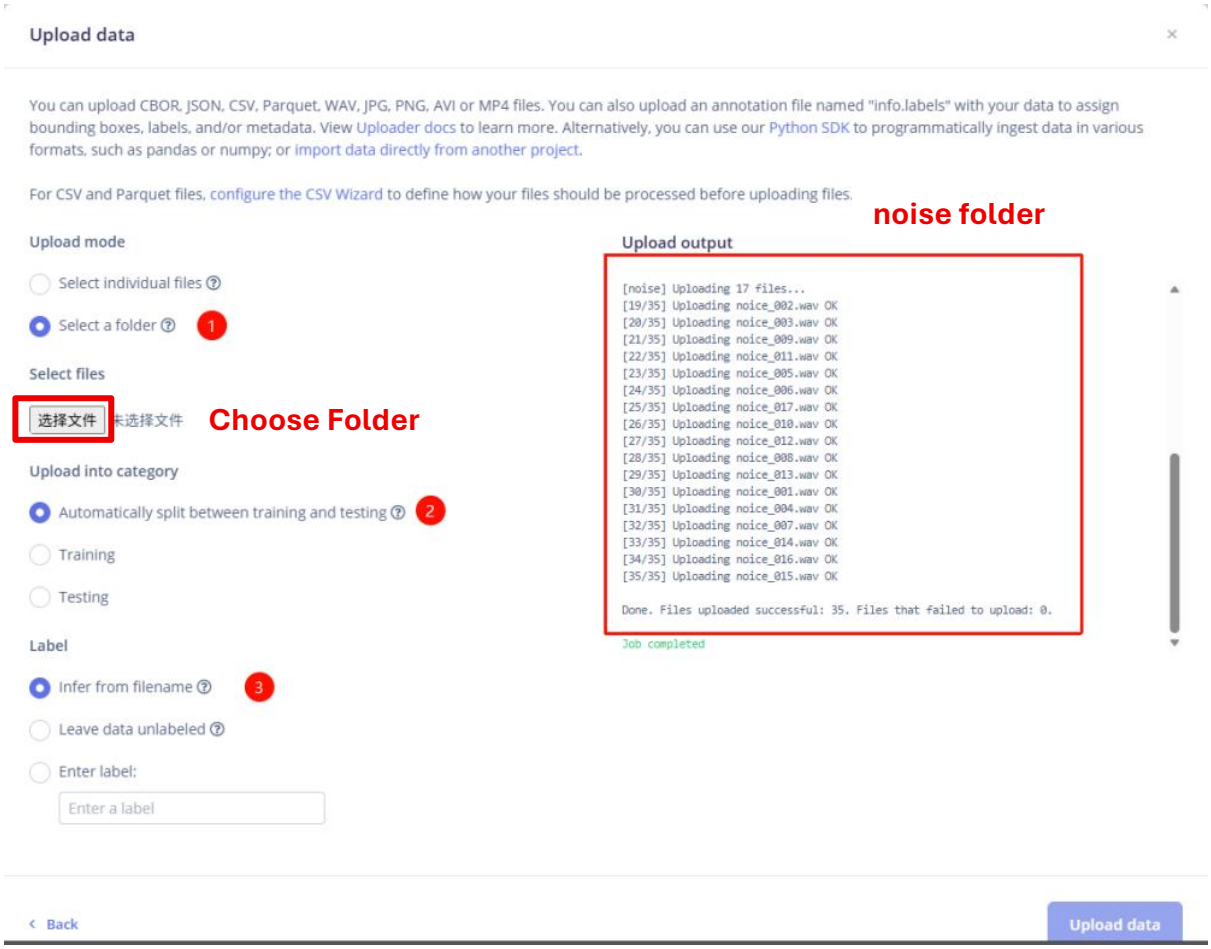
1. Click **add existing data**

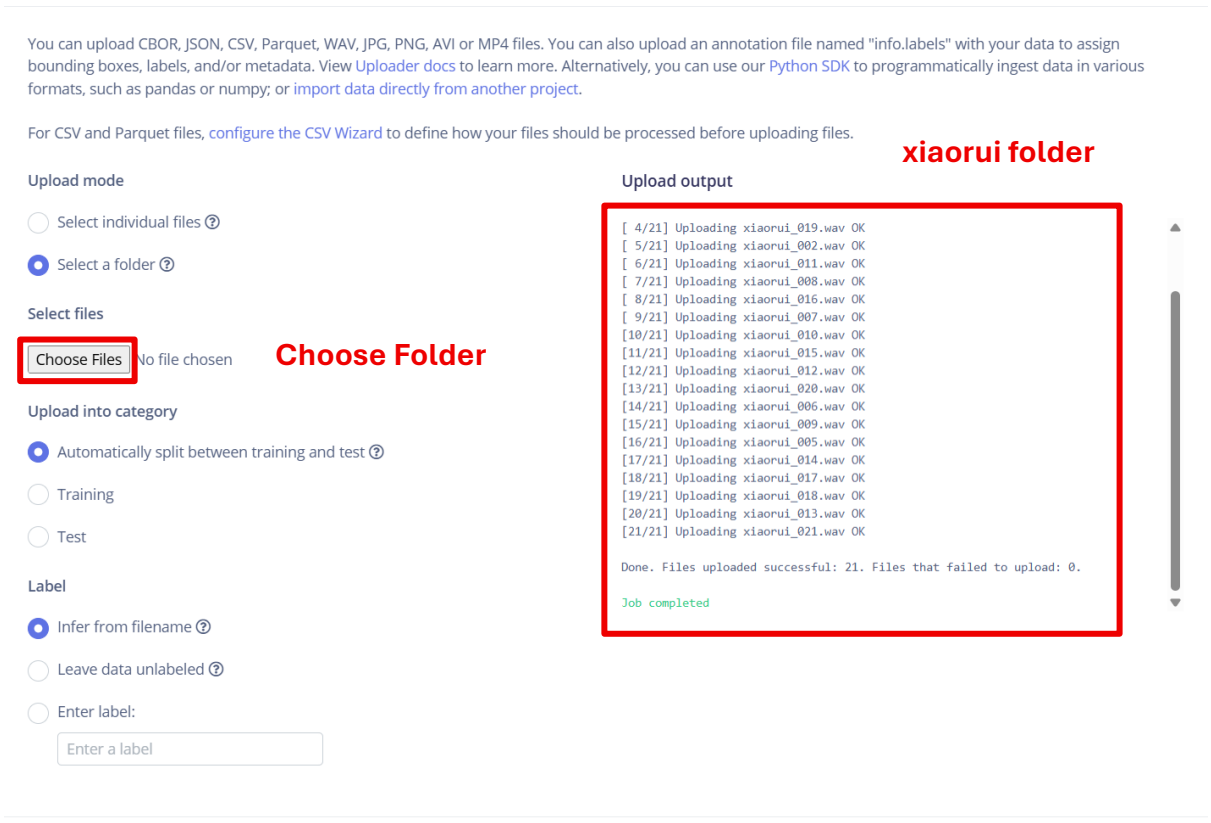
2. Select upload data





3. Select and upload the audio folder on your computer; the site will automatically classify the files based on their filenames (noise, wake word)

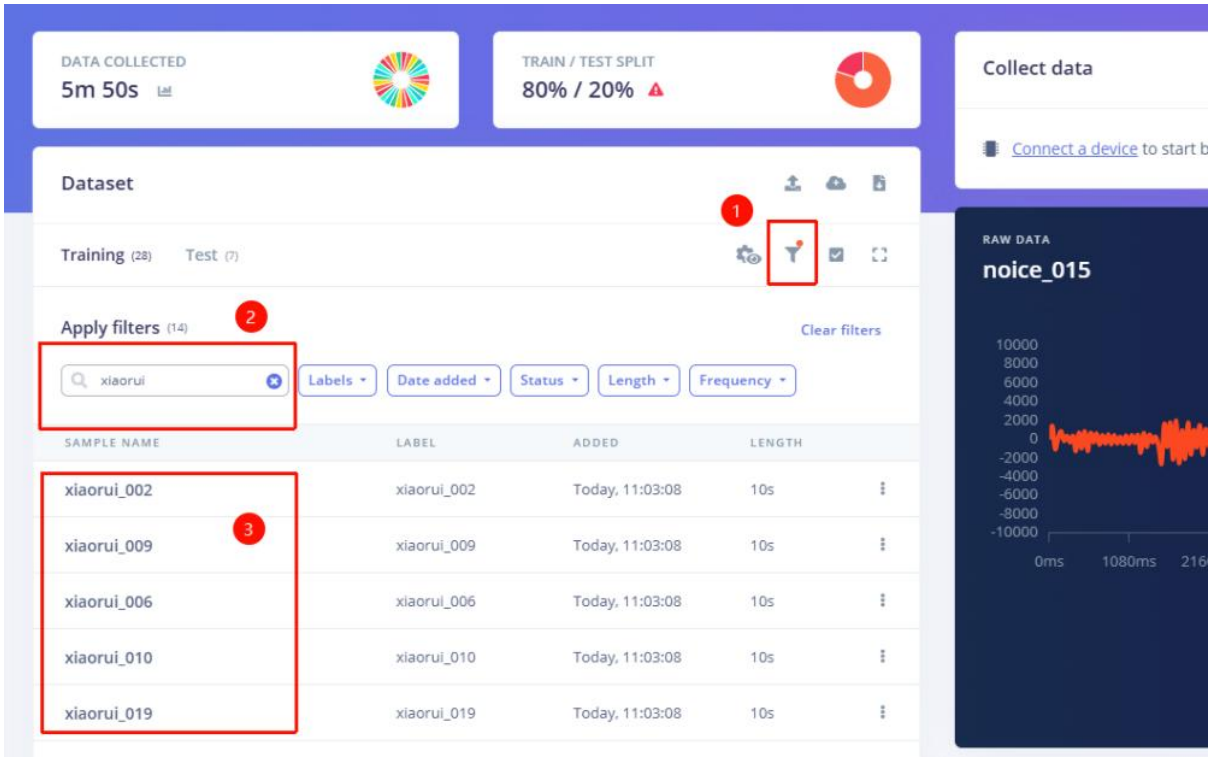

Upload data
You can upload CBOR, JSON, CSV, Parquet, WAV, JPG, PNG, AVI or MP4 files. You can also upload an annotation file named "info.labels" with your data to assign bounding boxes, labels, and/or metadata. View Uploader docs to learn more. Alternatively, you can use our Python SDK to programmatically ingest data in various formats, such as pandas or numpy; or import data directly from another project.
For CSV and Parquet files, configure the CSV Wizard to define how your files should be processed before uploading files.
noise folder
Upload mode
Select individual files
Select a folder
1
Select files
选择文件
未选择文件
Choose Folder
Upload into category
Automatically split between training and testing
2
Training
Testing
Label
Infer from filename
3
Leave data unlabeled
Enter label:
Enter a label
Upload output
[noise] Uploading 17 files...
[19/35] Uploading noice_002.wav OK
[20/35] Uploading noice_003.wav OK
[21/35] Uploading noice_009.wav OK
[22/35] Uploading noice_011.wav OK
[23/35] Uploading noice_005.wav OK
[24/35] Uploading noice_006.wav OK
[25/35] Uploading noice_017.wav OK
[26/35] Uploading noice_010.wav OK
[27/35] Uploading noice_012.wav OK
[28/35] Uploading noice_008.wav OK
[29/35] Uploading noice_013.wav OK
[30/35] Uploading noice_001.wav OK
[31/35] Uploading noice_004.wav OK
[32/35] Uploading noice_007.wav OK
[33/35] Uploading noice_014.wav OK
[34/35] Uploading noice_016.wav OK
[35/35] Uploading noice_015.wav OK
Done. Files uploaded successful: 35. Files that failed to upload: 0.
Job completed
Back
Upload data


You can upload CBOR, JSON, CSV, Parquet, WAV, JPG, PNG, AVI or MP4 files. You can also upload an annotation file named "info.labels" with your data to assign bounding boxes, labels, and/or metadata. View Uploader docs to learn more. Alternatively, you can use our Python SDK to programmatically ingest data in various formats, such as pandas or numpy; or import data directly from another project.
For CSV and Parquet files, configure the CSV Wizard to define how your files should be processed before uploading files.
xiaorui folder
Upload mode
Select individual files
Select a folder
Select files
Choose Files
No file chosen
Choose Folder
Upload into category
Automatically split between training and test
Training
Test
Label
Infer from filename
Leave data unlabeled
Enter label:
Enter a label
Upload output
[4/21] Uploading xiaorui_019.wav OK
[5/21] Uploading xiaorui_002.wav OK
[6/21] Uploading xiaorui_011.wav OK
[7/21] Uploading xiaorui_008.wav OK
[8/21] Uploading xiaorui_016.wav OK
[9/21] Uploading xiaorui_007.wav OK
[10/21] Uploading xiaorui_010.wav OK
[11/21] Uploading xiaorui_015.wav OK
[12/21] Uploading xiaorui_012.wav OK
[13/21] Uploading xiaorui_020.wav OK
[14/21] Uploading xiaorui_006.wav OK
[15/21] Uploading xiaorui_009.wav OK
[16/21] Uploading xiaorui_005.wav OK
[17/21] Uploading xiaorui_014.wav OK
[18/21] Uploading xiaorui_017.wav OK
[19/21] Uploading xiaorui_018.wav OK
[20/21] Uploading xiaorui_013.wav OK
[21/21] Uploading xiaorui_021.wav OK
Done. Files uploaded successful: 21. Files that failed to upload: 0.
Job completed

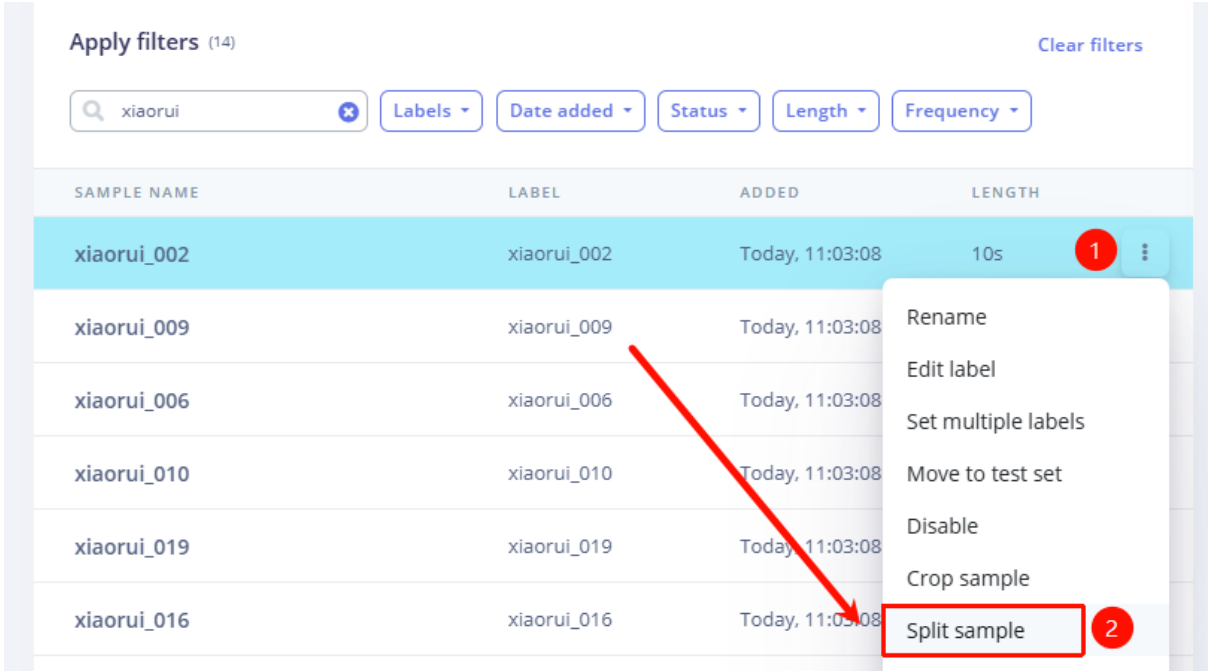
4 Data Splitting

Steps:

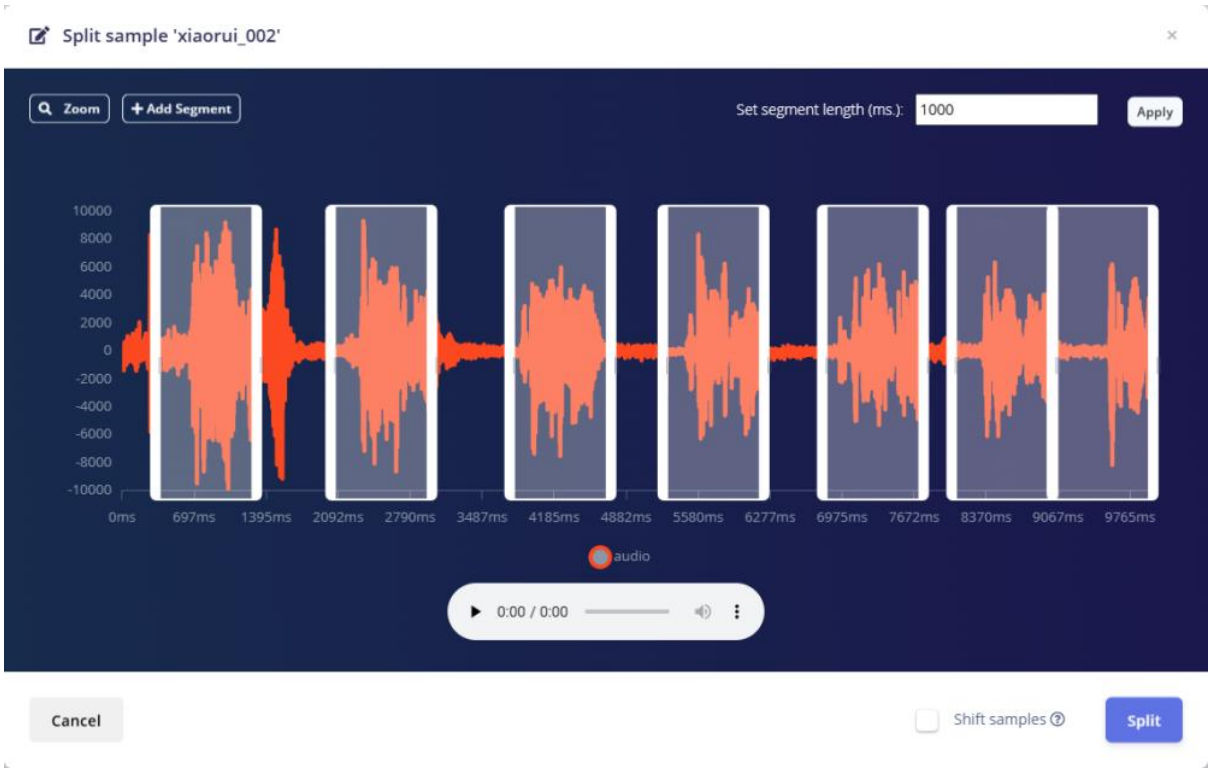
1. First select the Training list data, click Filter first, and enter our **wake word** file prefix to filter out the files we want to work with.



2. After clicking the audio file on the right, select **Split sample**, and the platform will split the audio file into roughly 1s audio segments

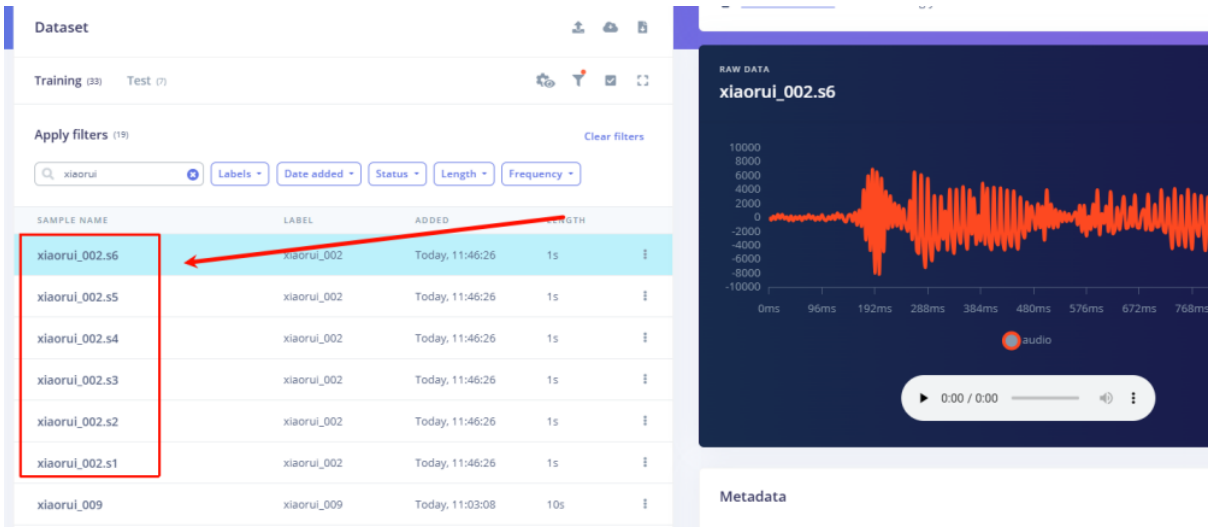


3. On the audio splitting screen, you can adjust the audio segments to cover the wake word audio content and play them back to listen and adjust.



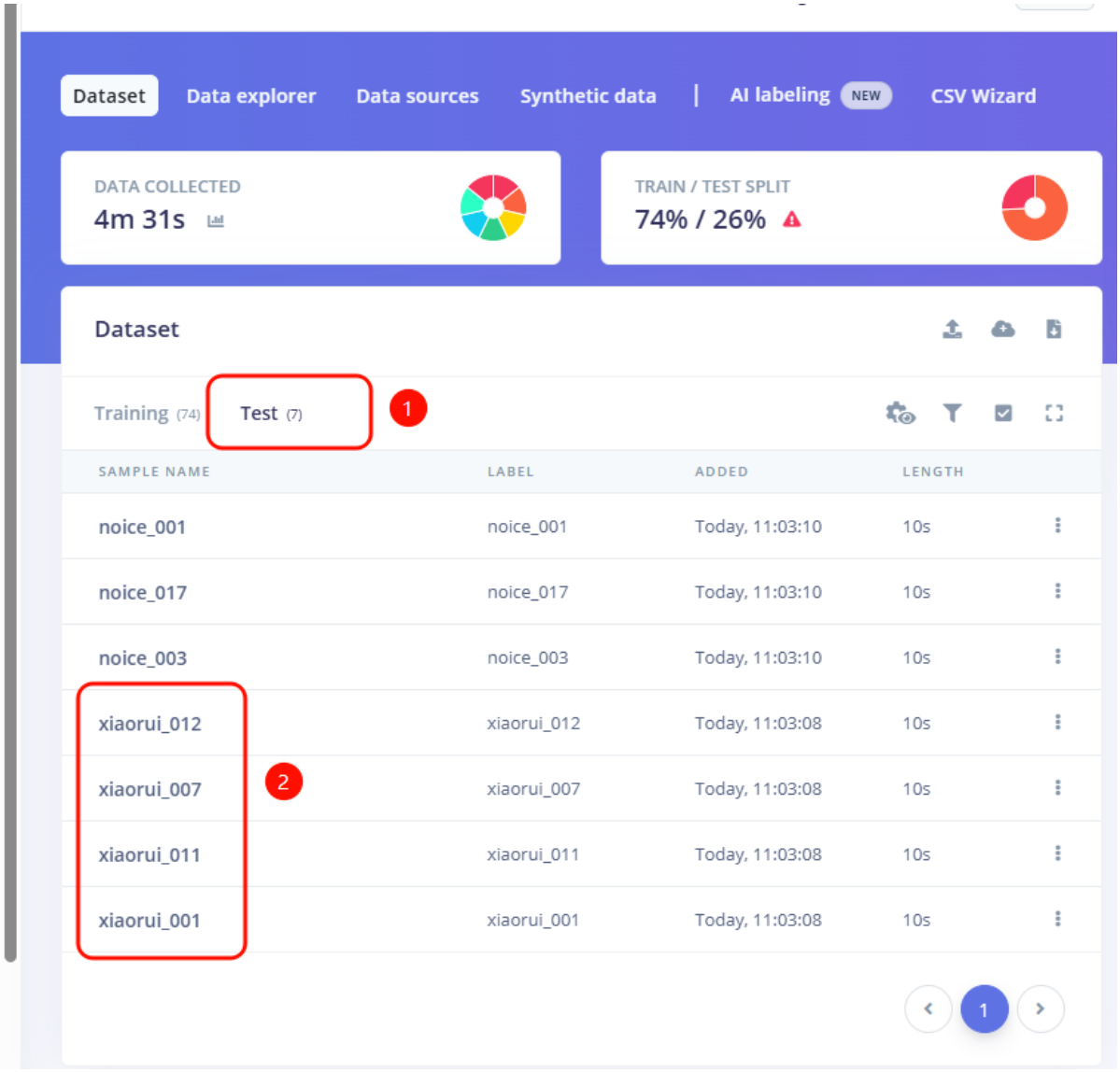
Each rectangle represents a sub-segment extracted by the split. If necessary, you can adjust the position of the rectangle so it fully covers our wake word audio area, or add/remove segments as needed.

4. After splitting, the audio files become independent files, and the site will automatically add a suffix (xxxx_Sn)

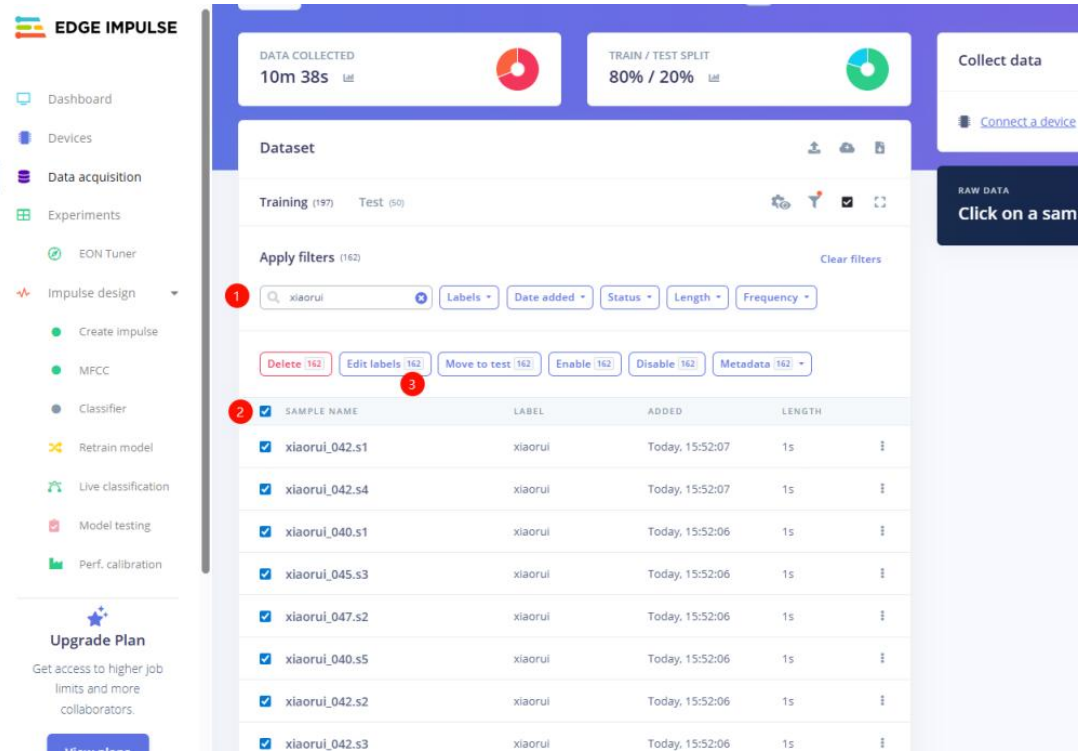


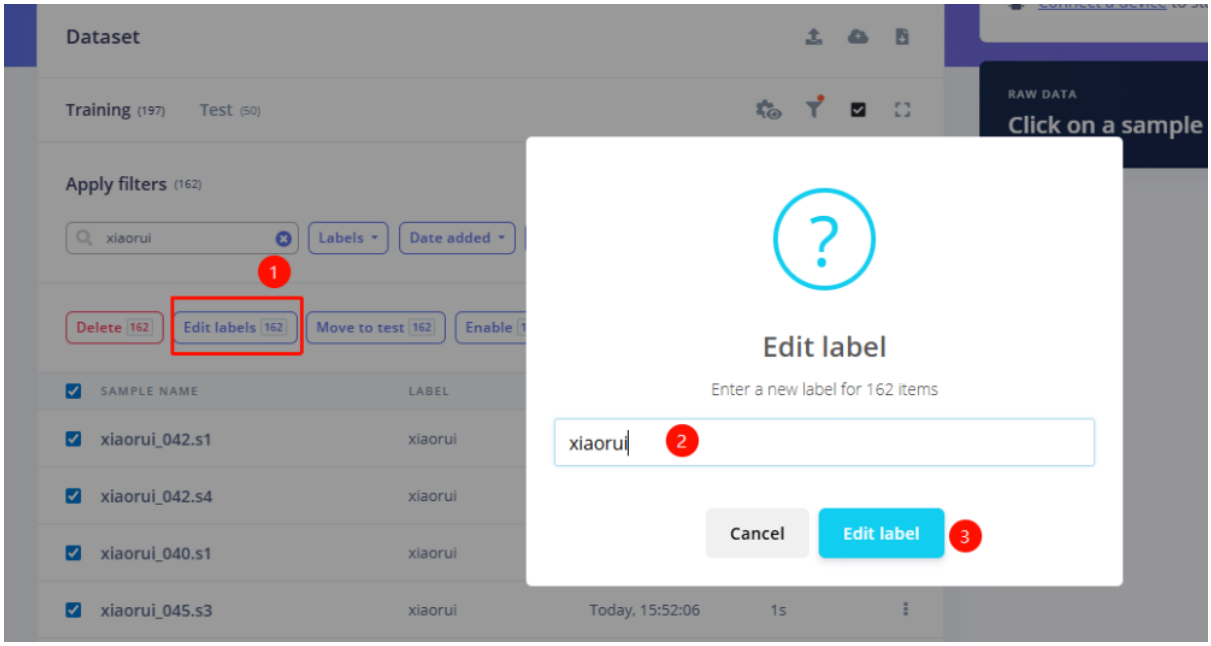
5. Following the steps above, split all the wake word files

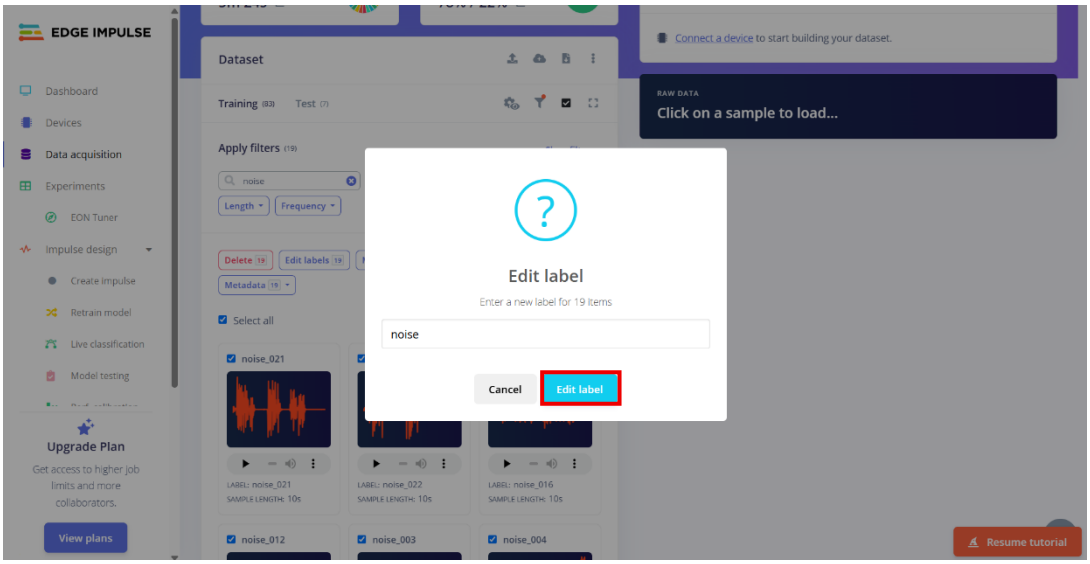
6. Select the Test list data; for the wake word files continue splitting; noise labelled files don't need to be split.

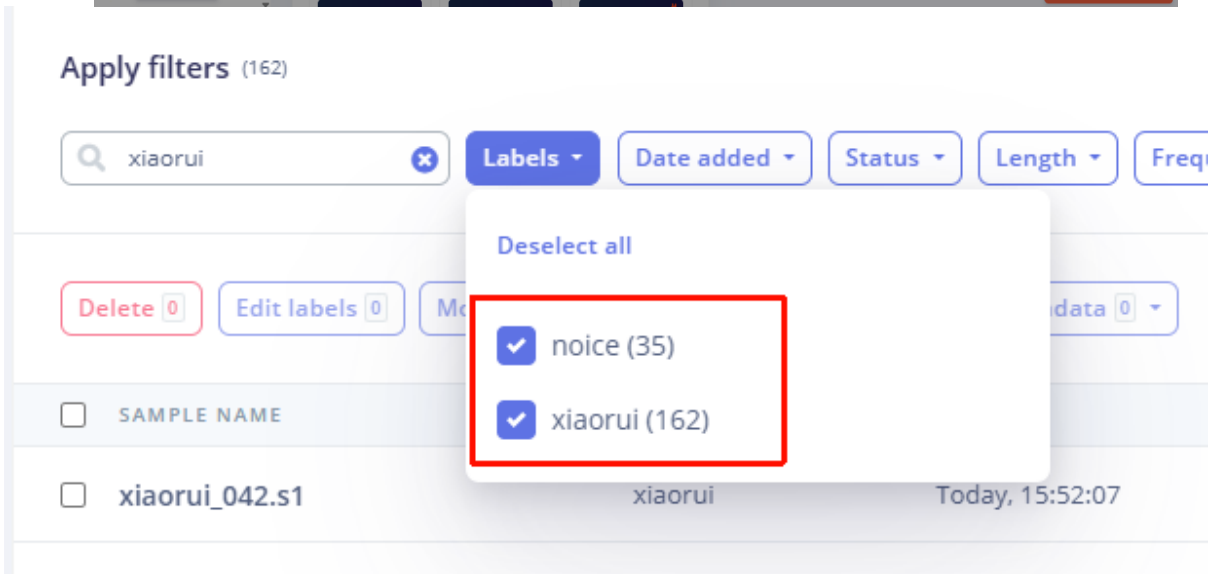


7. Classify the data: following the steps in the image below, divide the dataset into two main categories: Wake Word + Other.




Dataset
Training (197)
Test (50)
Apply filters (162)
xiaorui
Labels
Date added
1
Delete 162
Edit labels 162
Move to test 162
Enable
SAMPLE NAME
LABEL
xiaorui_042.s1
xiaorui
xiaorui_042.s4
xiaorui
xiaorui_040.s1
xiaorui
xiaorui_045.s3
xiaorui
Today, 15:52:06
1s
RAW DATA
Click on a sample
Edit label
Enter a new label for 162 items
xiaorui
2
Cancel
Edit label
3


EDGE IMPULSE
Dashboard
Devices
Data acquisition
Experiments
EON Tuner
Impulse design
Create impulse
Retrain model
Live classification
Model testing
Upgrade Plan
Get access to higher job limits and more collaborators.
View plans
Dataset
Training
Test
Apply filters
noise
Length
Frequency
Delete
Edit labels
Metadata
Select all
noise_021
noise_012
noise_003
noise_004
Connect a device to start building your dataset.
RAW DATA
Click on a sample to load...
Edit label
Enter a new label for 19 items
noise
Cancel
Edit label
Resume tutorial


Apply filters (162)
xiaorui
Labels
Date added
Status
Length
Deselect all
noice (35)
xiaorui (162)
Delete 0
Edit labels 0
SAMPLE NAME
xiaorui_042.s1
xiaorui
Today, 15:52:07

5 Model Training

5.1 Create Impulse (Preprocessing/Model Definition)

An "Impulse" here is the term used by **Edge Impulse** to describe the data processing—training pipeline.

Create the impulse and set the window size to 1000 ms, and set the window increase to 500 ms (overlapping windows to augment the data), with the frequency set to 16KHz.

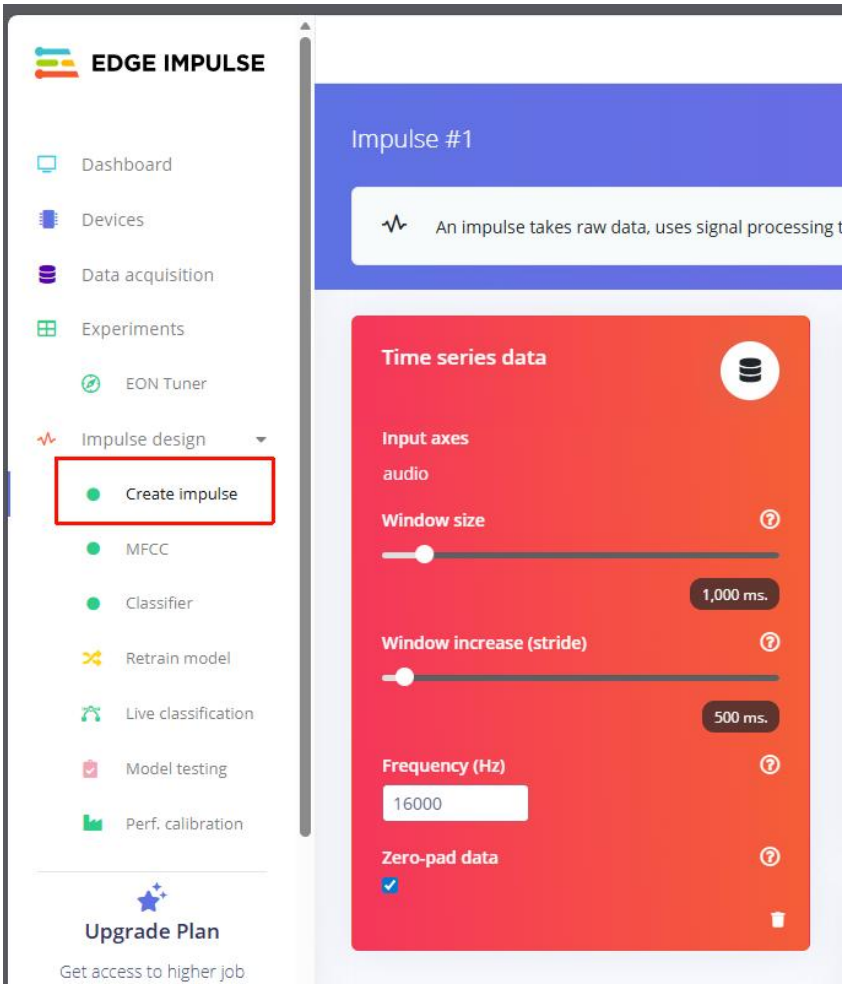
These settings mean that each time inference is run, sensor data collection will complete within 1000 ms, with the exact number of samples determined by the sampling frequency.

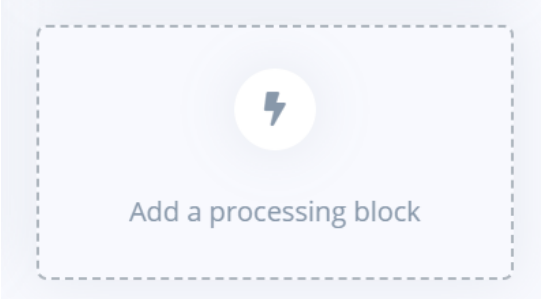
In short, your device will, within a 1000 ms time window, collect N data samples, then preprocess these samples and feed them into the neural network to obtain the inference result.

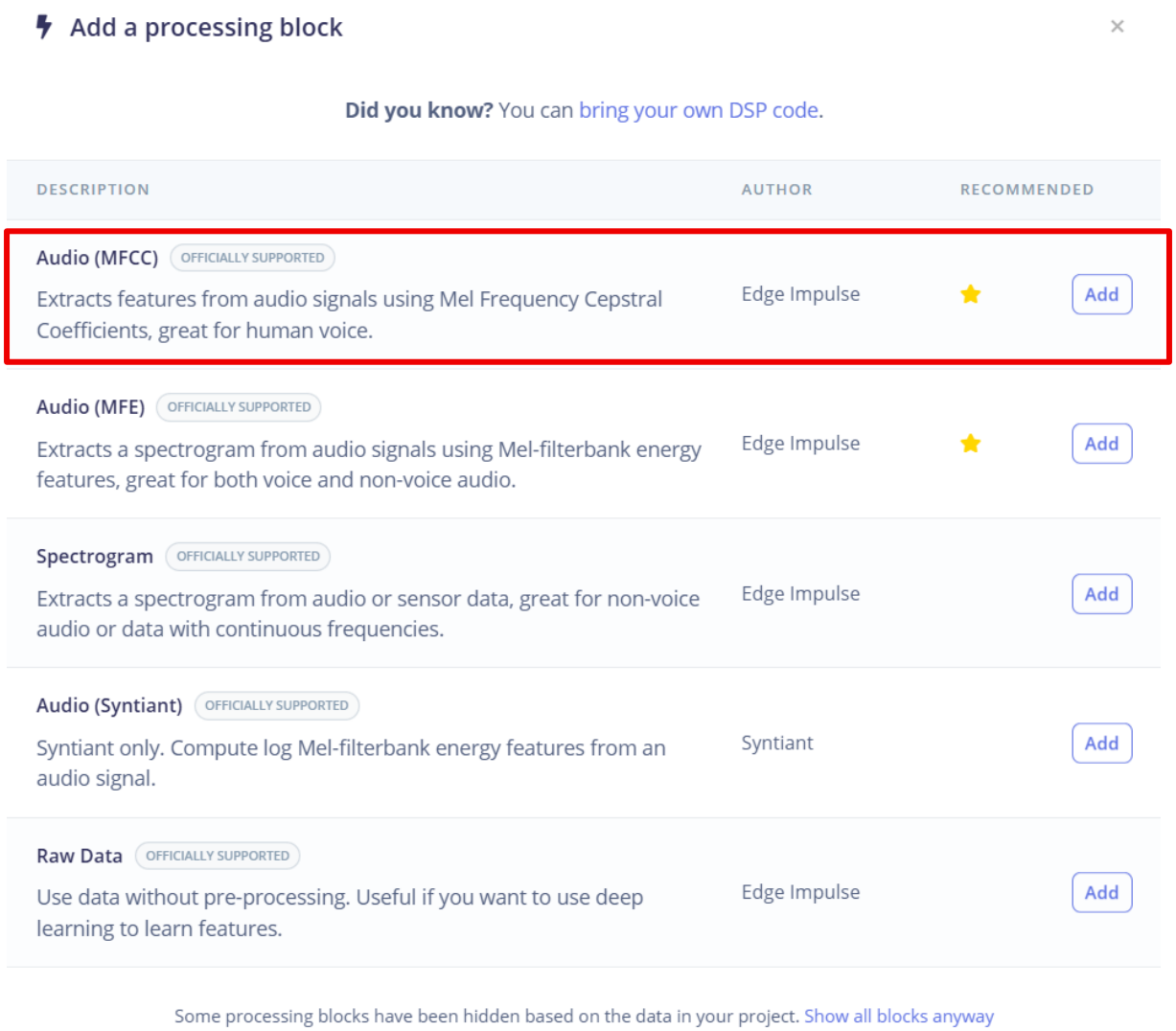
Steps:

1. Click on the left **Create impulse**

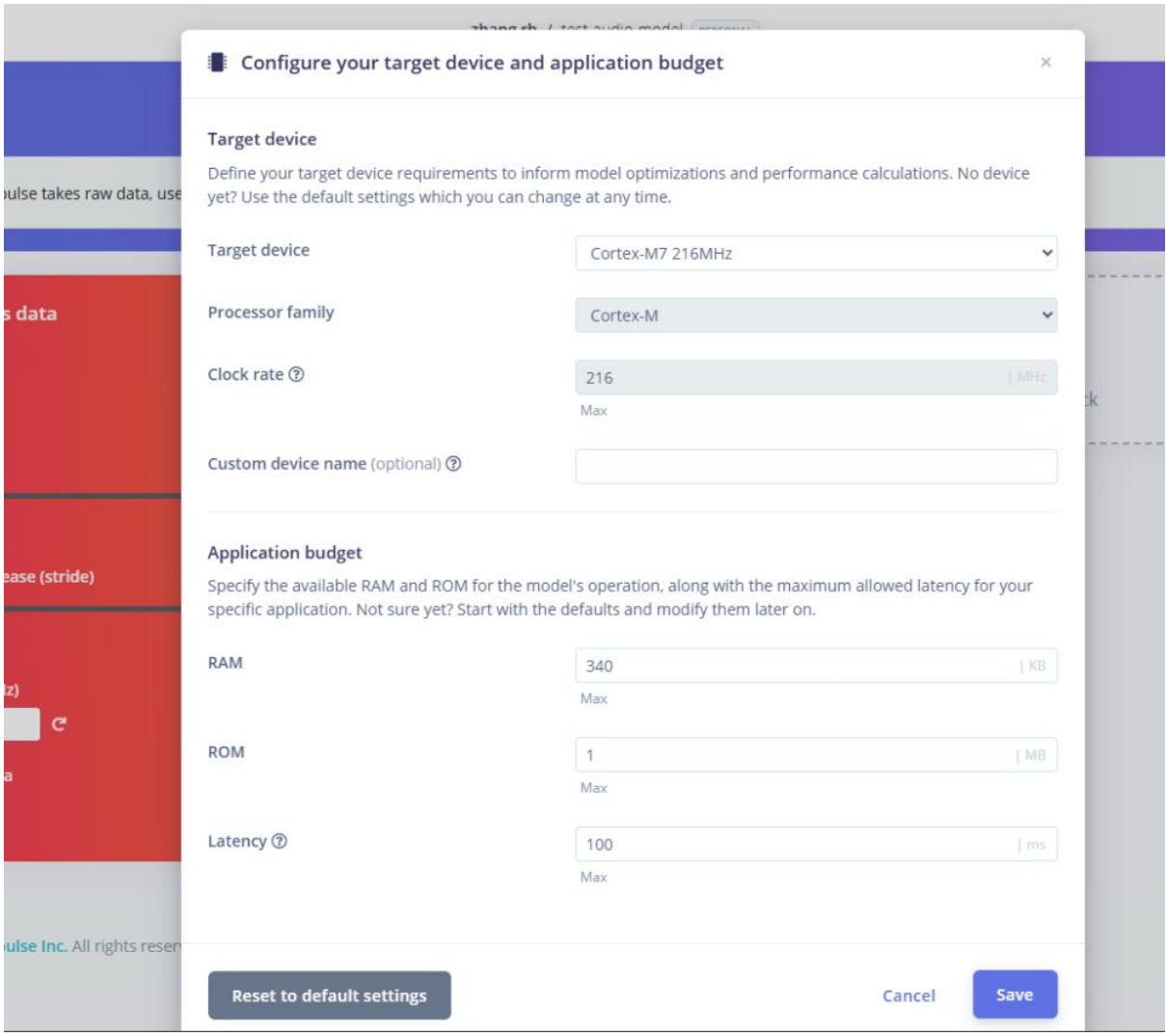
2. Then click **Add a processing block to** add Audio(MFCC)





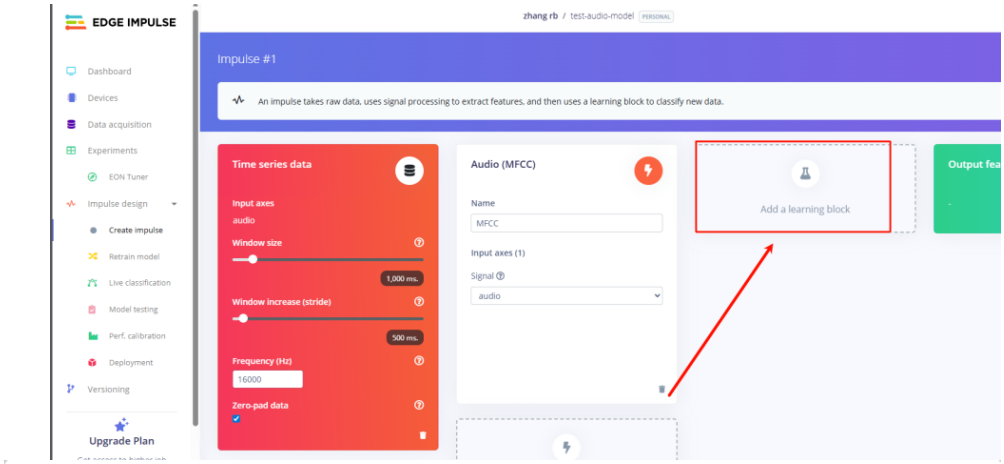


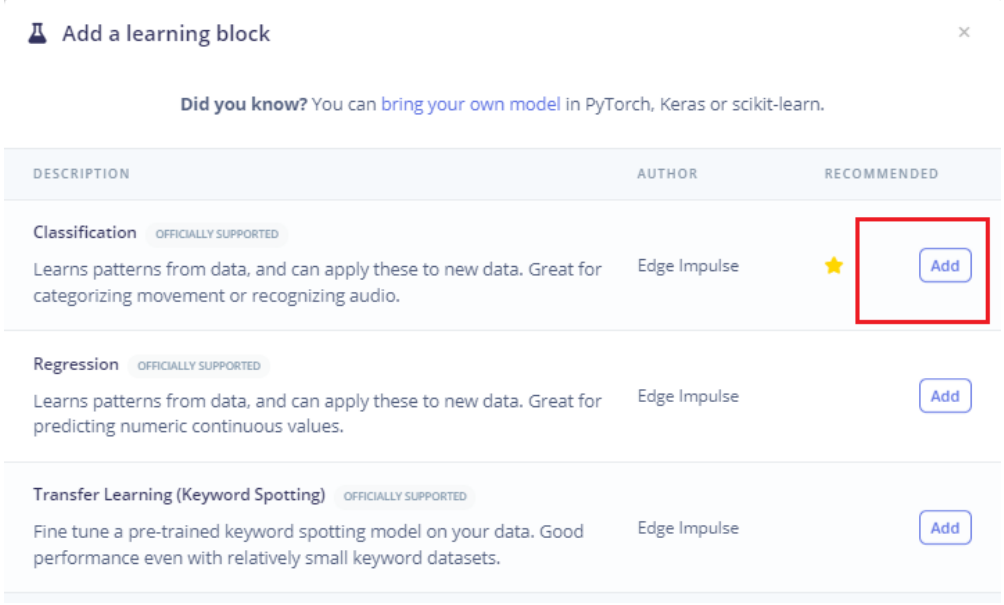
3. A dialog box will pop up below; the default configuration is fine:



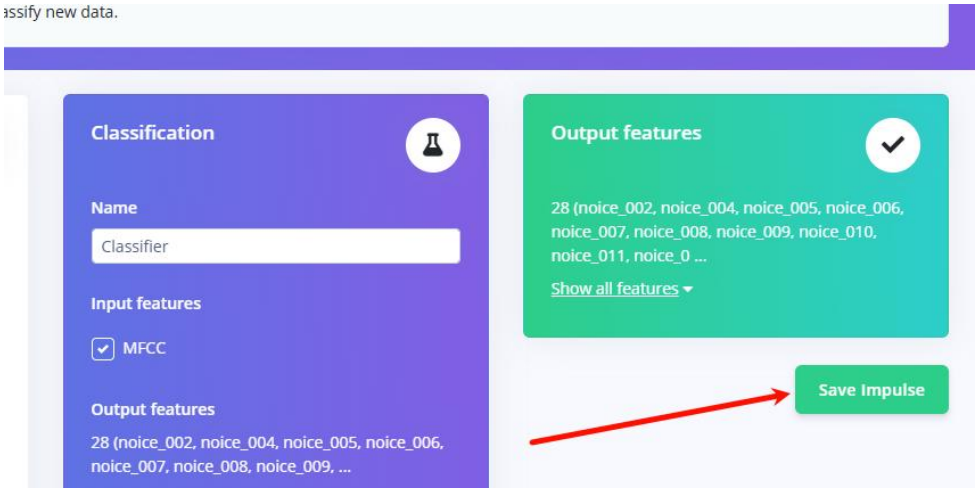
4. Use **MFCC**, which uses Mel-Frequency Cepstral Coefficients to extract features from the audio signal — very useful for human speech.

5. Then click **Add a learning block** to add the **Classification** block, which builds our model from scratch using a convolutional neural network for image classification.





6. Finally, click save impulse to save the configuration



5.2 Preprocessing (MFCC)

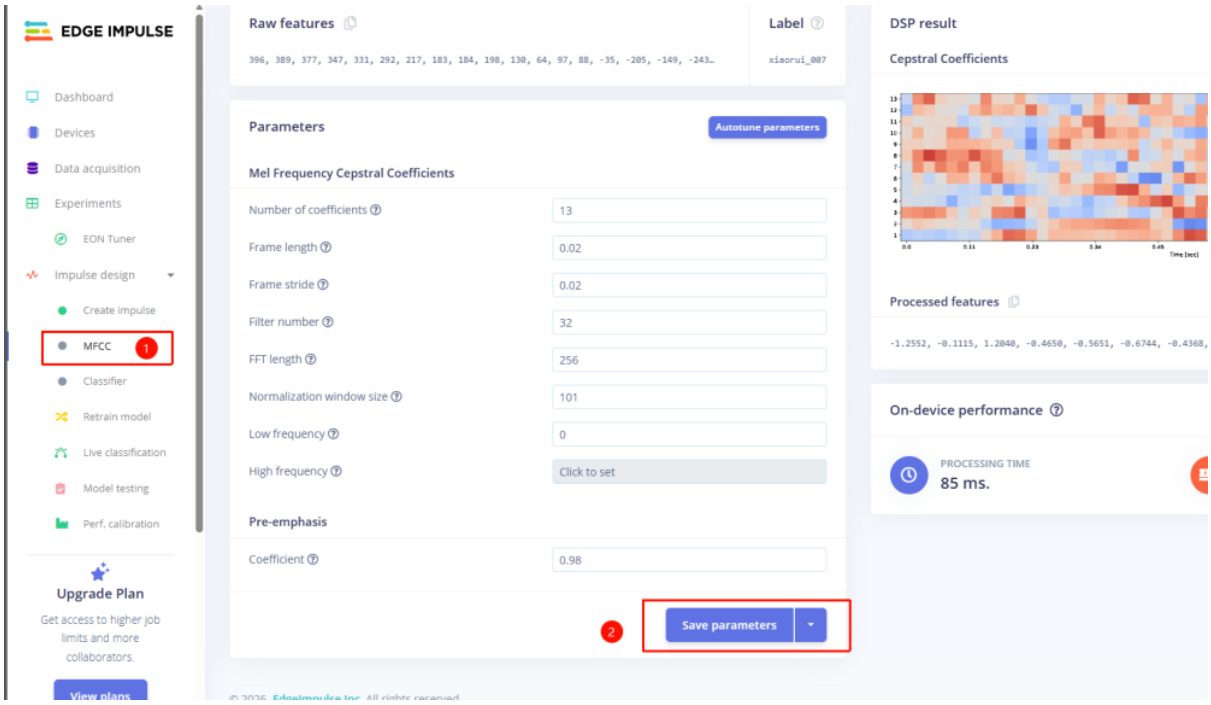
MFCC is a widely used method for converting audio signals into 2D features representing speech frequency patterns, which are well suited for speech-based recognition models.

We create spectrogram images from the recorded audio.

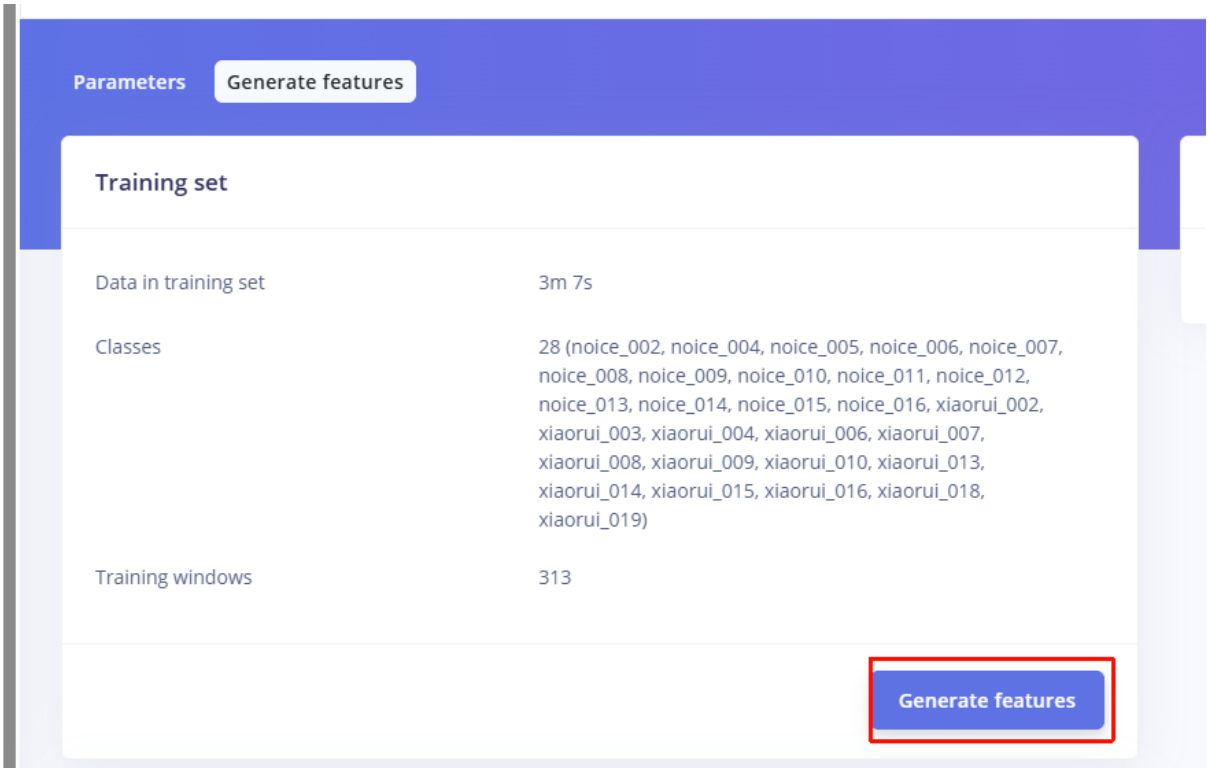
Steps:

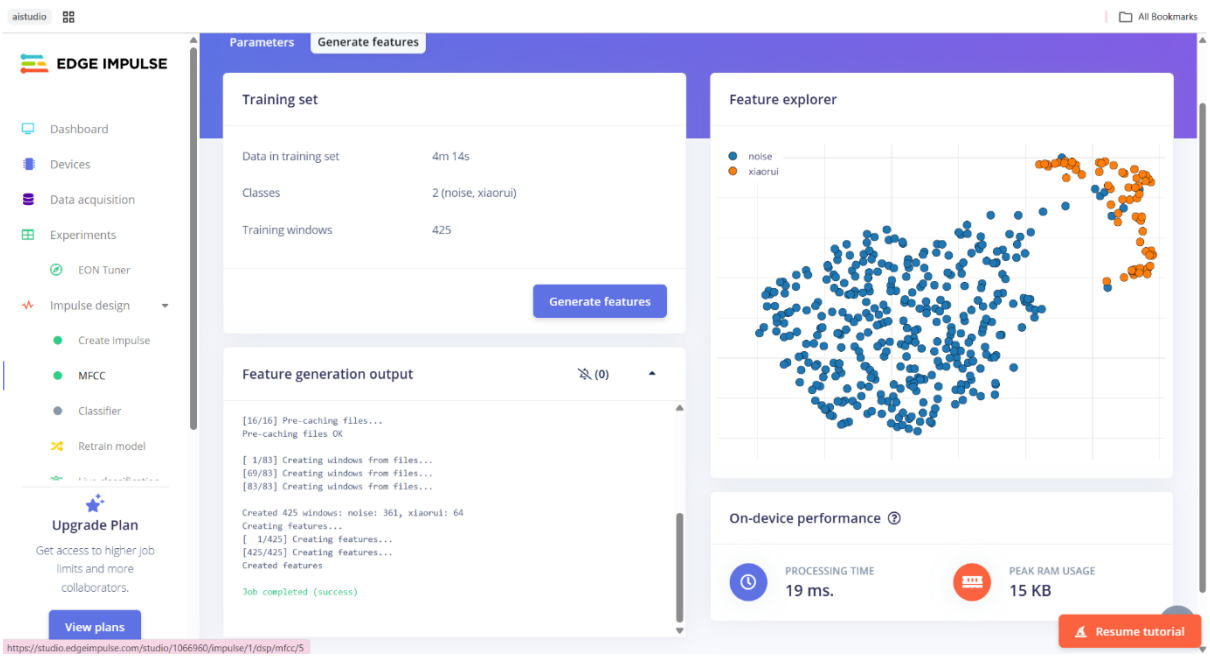
1. Click **MFCC**, we can keep the default parameter values.

2. Click **Save parameters**.



3. Click **Generate Features** to generate 3 labels' worth of feature data.


Parameters
Generate features
Training set
Data in training set
3m 7s
Classes
28 (noice_002, noice_004, noice_005, noice_006, noice_007, noice_008, noice_009, noice_010, noice_011, noice_012, noice_013, noice_014, noice_015, noice_016, xiaorui_002, xiaorui_003, xiaorui_004, xiaorui_006, xiaorui_007, xiaorui_008, xiaorui_009, xiaorui_010, xiaorui_013, xiaorui_014, xiaorui_015, xiaorui_016, xiaorui_018, xiaorui_019)
Training windows
313
Generate features


EDGE IMPULSE
Dashboard
Devices
Data acquisition
Experiments
EON Tuner
Impulse design
Create impulse
MFCC
Classifier
Retrain model
Upgrade Plan
Get access to higher job limits and more collaborators.
View plans
Parameters
Generate features
Training set
Data in training set
4m 14s
Classes
2 (noise, xiaorui)
Training windows
425
Generate features
Feature generation output
[16/16] Pre-caching files...
Pre-caching files OK
[1/83] Creating windows from files...
[69/83] Creating windows from files...
[83/83] Creating windows from files...
Created 425 windows: noise: 361, xiaorui: 64
Creating features...
[1/425] Creating features...
[425/425] Creating features...
Created features
Job completed (success)
Feature explorer
noise
xiaorui
On-device performance
PROCESSING TIME
19 ms.
PEAK RAM USAGE
15 KB
Resume tutorial

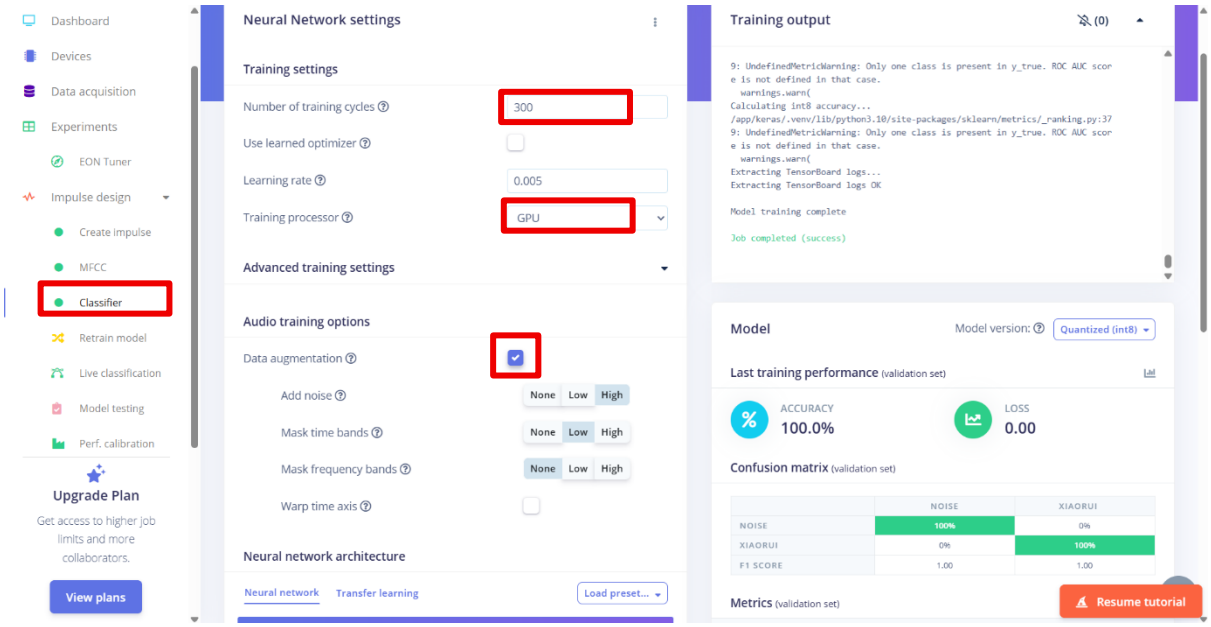
5.3 Model Design and Training (Classifier)

Next, we need to design the model structure and begin training. Steps are as follows:

Steps:

1. Click on the left **Classifier**, the entire model structure design has already been configured.

2. Then click **Save & Train** to start training the model.



Through multiple iterations, model generalization can be improved.

Verified improvement methods include: optimizing the test dataset, adjusting the model structure(e.g., adding more convolutional layers), etc.

The platform has training time limits; free users can only train for up to 20 minutes;

If unlimited training time is needed, a paid service must be purchased.

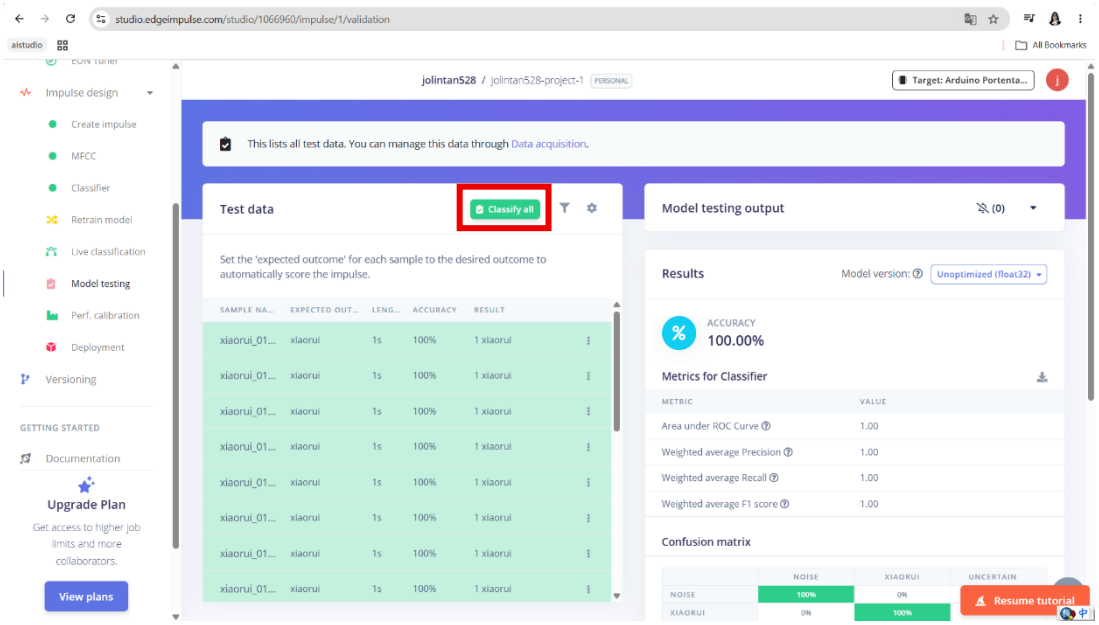
6 Model Evaluation

Steps:

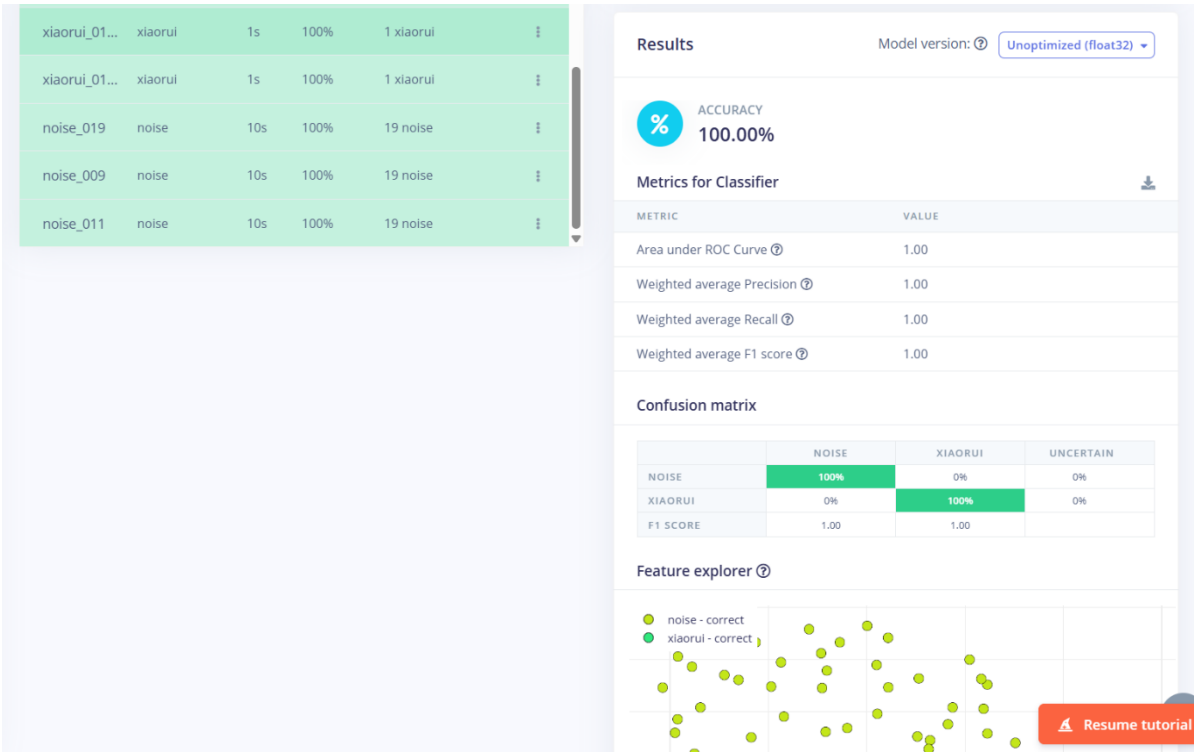
1. Click on the left **Model testing.**

2. Then click **Classify all.**

3. Begin classifying all the test set data.



4. After testing, you can view the test results in the **Result** panel on the right.



7 Model Integration

7.1 Generate Model Library File

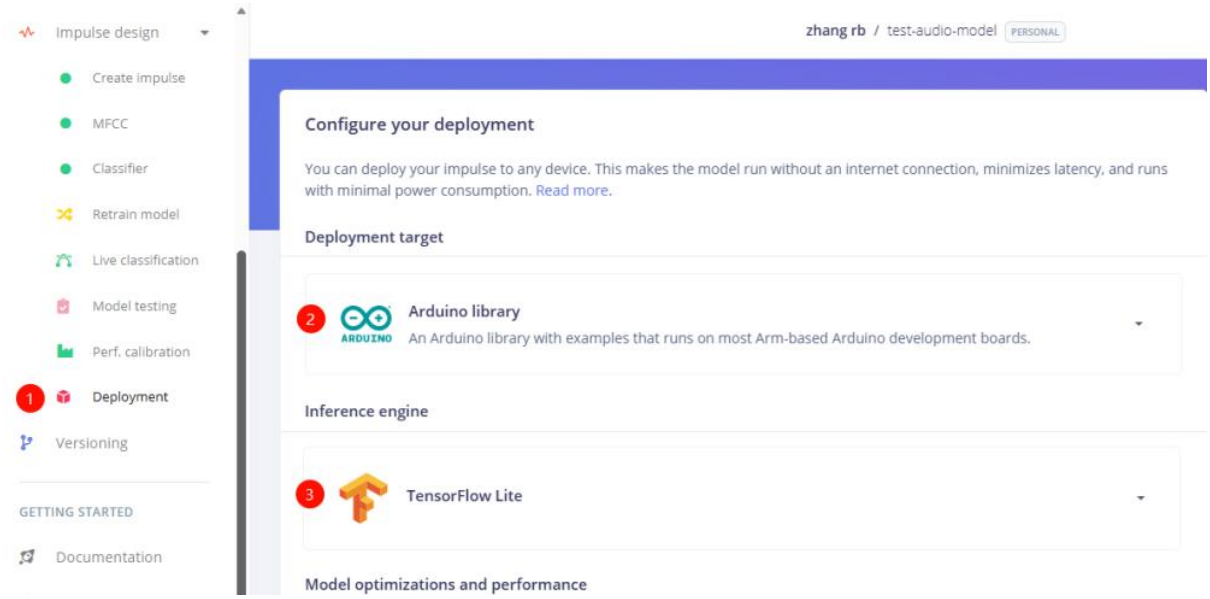
After the model training is complete, we need to generate a library file that runs on the Edgi-Talk platform.

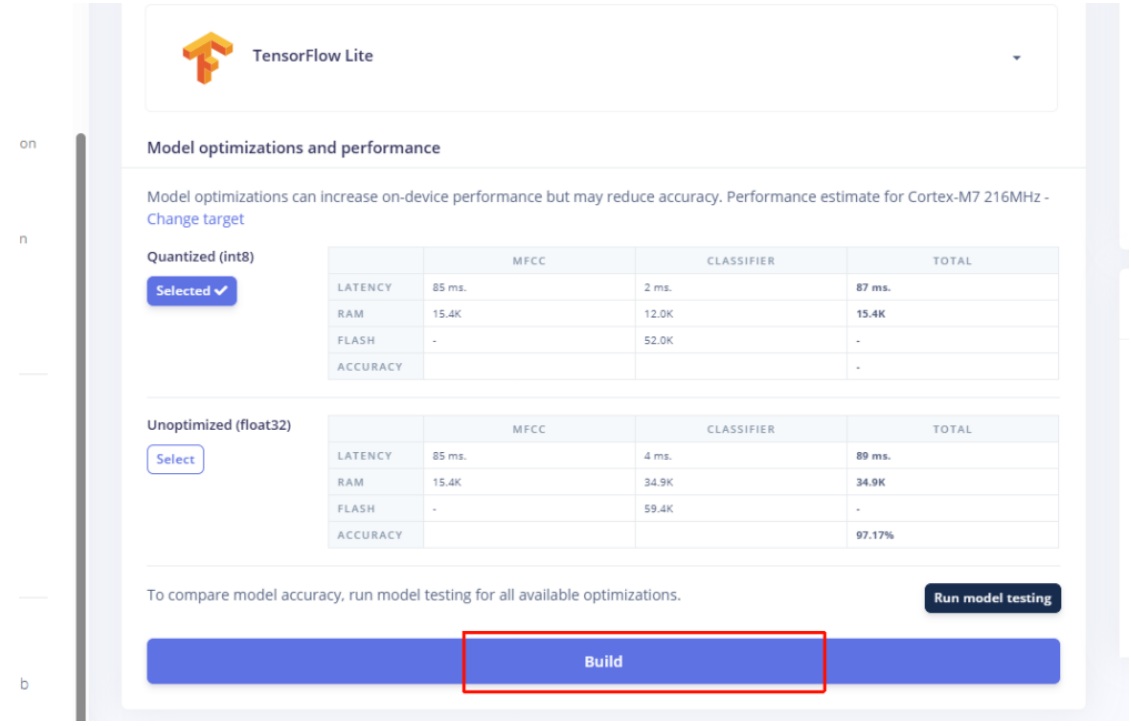
Steps:

1. Click on the left Deployment.

2. Search for Arduino library and TensorFlow Lite.

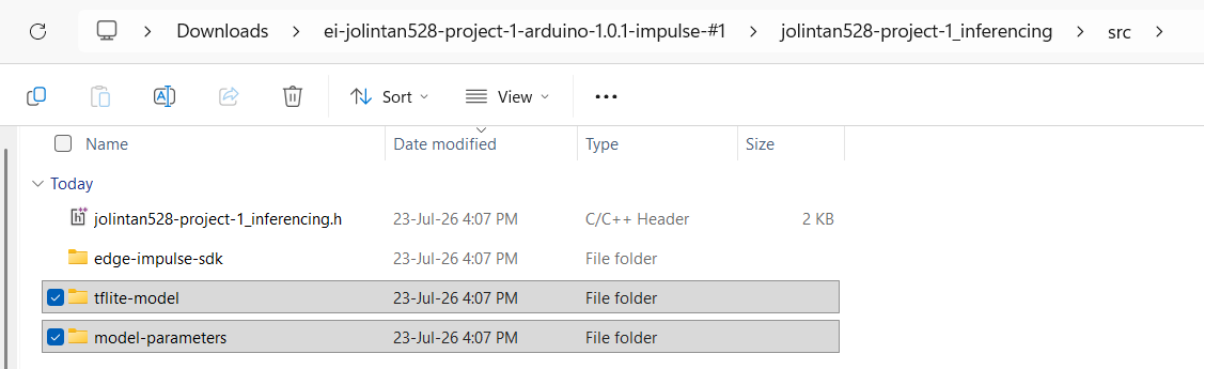
3. Then click Build, and once the build is complete, save the downloaded library file.



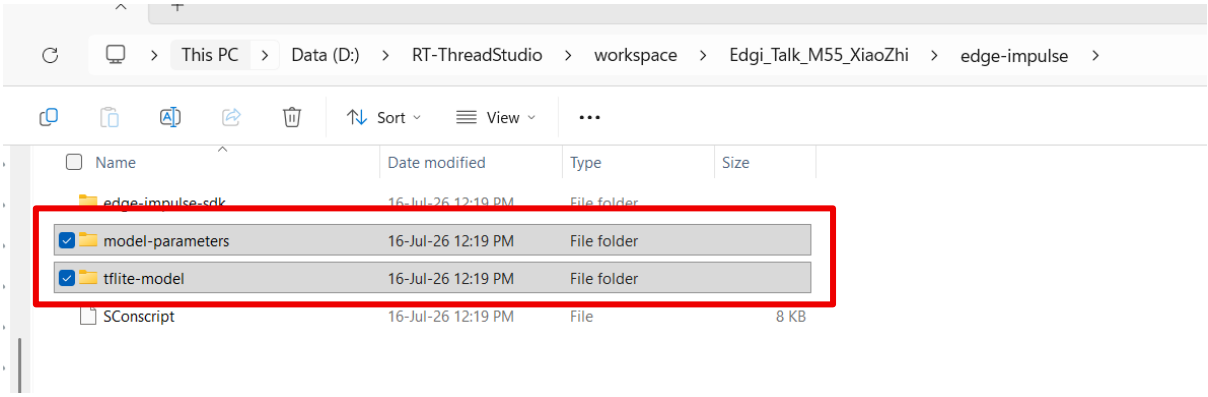


7.2 Deploy to Embedded Hardware

1. Extract the zip file downloaded from the previous step to get the following file directory.



2. Get the latest Edgi-Talk dev board SDK, go into the Edgi_Talk_M55_XiaoZhi/edge-impulse directory, delete the two folders shown in the image below, then replace them with the two folders from the previous step to replace the model files.



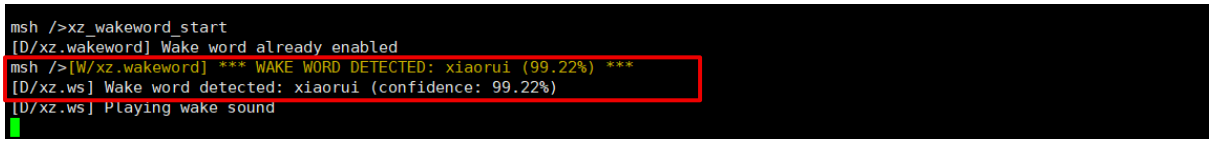
3. Compile the Edgi_Talk_M55_XiaoZhi project and flash it onto the dev board.

4. In the Edgi-Talk serial terminal, enter the following command to start the wake word test:

xz_wakeword_init

xz_wakeword_start

5. Say the wake word to the dev board; the demo here uses "xiaorui "; when the wake word is spoken, the serial terminal will print a log



Note: If cannot type anything into the terminal, try

1. Opened the M33 Blink LED project's settings

- Expanded Edgi_Talk_M33_Blink_LED in Project Explorer → opened **RT-Thread Settings**.

2. Enabled the CM55 core

- Went to **Hardware** tab → **SOC Multi Core Mode** → checked **"Enable CM55 Core"**.
- This is the critical setting — without it, the M33 firmware boots but never hands off execution to the M55 core, so your already-flashed XiaoZhi app in external flash just sits there unused.

3. Built the M33 project

- Set it as active project → built it → got a fresh rtthread.hex in Edgi_Talk_M33_Blink_LED\Debug\.

4. Flashed it via ModusToolbox Programmer

- Loaded that new M33 hex file (same Programmer GUI settings as before: External Memory ticked, PSE84_SMIF.FLM flashloader).
- Clicked Connect → Program. This wrote to the **internal flash bank** (cat1d.cm33.main_ns, 0x22... region) — the part that was never touched when you'd only flashed M55 before.

5. Did NOT need to reflash XiaoZhi again

- Because M33 and M55 live in separate flash regions (internal vs. external QSPI), your earlier XiaoZhi flash was still intact. Flashing M33 with CM55 enabled just gave it the "green light" to boot into the M55 app that was already sitting there.

6. Reset the board

- Power-cycled / reset → this time the boot sequence ran correctly: Secure M33 → M33 → **M55 started**, XiaoZhi UI initialized, msh /> prompt appeared.